

基本概念

缩进

- 行开头处的空白被用来计算该行的缩进等级，以决定语句段落的组织结构。
- 缩进不仅仅是为了美观和可读性，而且是Python语法的一部分，不正确的缩进可能导致语法错误或逻辑错误。
- Python中的复合语句（如条件语句、循环循环、函数定义等）通常需要使用缩进来表示代码块的层次结构。一次缩进通常是4个空格。

```
def add(left, right):  
    print(left + right)  
    print(left * right)
```

```
add(1, 2)
```

```
add(3, 4)
```

注释

- 注释在 Python 解释器的语法分析中会被忽略。通俗的说，注释是给人看的，而程序执行时会将它无视。
- 单行注释：以 # 开头定义。
- 多行注释：用成对的三个引号定义。

```
# 定义一个求减法的函数
```

```
def sub(left, right): # left和right是形参  
    print(left - right)  
    print(right - left)
```

```
'''
```

上面的函数只是做了定义，
还没有被调用，所以不会执行函数体

```
'''
```

```
"""
```

上面的函数没有返回值，
或者可以理解为return None

```
"""
```

拼接

- 显式的行拼接

如果某行代码过于冗长，可以用续行符（反斜杠\）将一行代码分成多行书写，以提高代码的可读性和清晰度。

```
""" 下面 a, b 两种写法是等价的 """
```

```
a = 1 + 2 + 3 + 4 + 5 + 6
```

```
b = 1 + 2 + \  
    3 + 4 + \  
    5 + 6
```

- 隐式的行拼接

列表，元组，字典和集合，无需使用续行符，它们的元素会自动拼接为一行。

```
""" 下面 num1, num2 两种写法是等价的 """
```

```
num1 = [1, 2, 3,  
        4, 5, 6,  
        7, 8, 9]
```

```
num2 = [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

变量

- 变量就是可以变化的量，是计算机语言中能表示值的抽象概念。

- 变量需要先定义再使用，解释器执行到变量定义的代码时会申请内存空间存放数据，然后将数据所在的内存地址绑定给变量名。
- 变量的定义由三部分组成：
 - a. 变量名：指向数据所在的内存地址，通过它访问数据。
 - b. 等号（=）：赋值符号，用来将数据所在的内存地址绑定给变量名。
 - c. 值：内存地址对应空间存储的数据。
- 标识符：用于变量、函数、类、模块等的名称。
- 变量命名规范（其它标识符也是类似）：
 - a. 变量名通常只包括字母、数字和下划线，且不能以数字开头。
 - b. 变量名是区分大小写的，如：`num` 和 `Num` 是两个不同的变量名。
 - c. 由多个单词构成的变量名，可用下划线来分隔这些单词，或使用驼峰命名法。
 - d. 变量名要尽量简短及具描述性，尽量做到短小精悍，见名知义。
 - e. 不能使用关键字作为变量名。
 - f. 尽量不要使用内置函数名作为变量名。

输出 & 输入

输出

`print(*values, sep=' ', end='\n')`

- **values**: 要输出的对象，可以是多个对象
- **sep**: 输出的多个对象用什么间隔，默认为一个空格字符
- **end**: 输出最后用什么结尾，默认为一个换行符

```
print('hello world')  
print(1, 2, 3, sep='-', end='-456')
```

输入

`input([prompt])`

- **prompt**: 提示信息
- 接收控制台输入的数据，以字符串的形式返回该数据

```
name = input('请输入你的姓名: ')  
print(name, '你好，很高兴认识你!')
```